

Министерство просвещения Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, физики, информатики и технологий  
Кафедра информатики, информационных технологий  
и методики обучения информатике

# МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОСТРОЕНИЯ ПРОСТЕЙШИХ ГРАФИКОВ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ CANVAS

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
09.03.02 – Информационные системы и технологии*

Исполнитель: студент группы ИСиТ-1601  
Института математики, физики,  
информатики и технологий  
Токарева Н.А.

Допустить к защите  
«\_\_\_» \_\_\_\_\_ 2020 г.

Зав. кафедрой \_\_\_\_\_  
М.В. Лапенков

Руководитель: старший преподаватель  
кафедры ИИТиМОИ  
Старкова Л.Н.

Руководитель ОПОП  
\_\_\_\_\_  
Л.В. Сардак

Екатеринбург – 2020

## Реферат

Токарева Н.А. МОБИЛЬНОЕ ПРИЛОЖЕНИЕ ДЛЯ ПОСТРОЕНИЯ ПРОСТЕЙШИХ ГРАФИКОВ ФУНКЦИЙ С ИСПОЛЬЗОВАНИЕМ CANVAS, выпускная квалификационная работа: 50 стр., рис. 22, библи. 34, приложений \_.

Ключевые слова: образование, мобильное приложение, информационная система, фреймворк Ionic, график функций.

Объект разработки: мобильное приложение для построения простейших графиков функций с использованием Canvas.

Цель разработки: спроектировать и разработать мобильное приложение для построения простейших графиков функций с использованием Canvas.

В рамках выпускной квалификационной работы проведен анализ достоинств и недостатков внедрения мобильных устройств в процесс обучения, доказано преобладание положительных качеств мобильного обучения. Изучены и выбраны возможные технологии и инструменты написания программы. Реализовано приложение построения графиков функций в соответствии с техническим заданием.

## Оглавление

Введение .....	4
Глава 1. Мобильные технологий в образовательном процессе .....	6
1.1. Использование мобильных устройств в обучении .....	6
1.2. Выбор инструментов для написания мобильного приложения ..	11
1.3. Формализованное описание технического задания .....	22
Глава 2. Разработка мобильного приложения .....	26
2.1. Модельное представление мобильного приложения.....	26
2.2. Описание приложения.....	31
2.3. Сопроводительная документация .....	39
Заключение .....	45
Список информационных источников .....	46
Приложения .....	50

## Введение

Человеку всегда было характерно желание упростить себе жизнь, как можно больше минимизировать свои действия для выполнения максимального количества задач. Как следствие, в настоящее время большинство предприятий и организаций достаточно автоматизированы, что позволяет человеку лишь контролировать работу механизмов без необходимости вручную выполнять все действия самостоятельно. В качестве примера подобного проявления автоматизации можно рассматривать современные мобильные устройства, в частности мобильные технологии, развитие которых за последние годы продвинулось далеко вперед.

Практически у каждого современного человека есть смартфон, который некоторое время назад считался лишь средством связи, но сейчас развитие мобильных технологий позволяет работать и/или учиться из любого удобного для пользователя места, с минимальными затратами, все что необходимо – приложение или выход в Интернет.

Смотря на статистику [1], самыми активными пользователями мобильных приложений являются люди, возраст которых от 12 до 24, поэтому неудивителен спрос на приложения и сервисы, помогающие в образовании [2]. На сегодняшний день крупнейшими поставщиками мобильных приложений являются Play Маркет у Android, и App Store у iOS. Если брать за основной источник приложений – Play Маркет и смотреть на технический инструментарий, можно найти различные калькуляторы и таблицы, но очень мало приложений для построения математических графиков, которые могут быть весьма полезны не только для изучения самих графиков в школах, но и для графического представления данных в любых других сферах.

**Предмет разработки:** мобильное приложение для построения простейших графиков функций с использованием элемента HTML5 Canvas.

**Цель разработки:** построение простейших графиков функций с применением элемента HTML5 Canvas.

**Задачи:**

1. Произвести терминологический анализ, выявить проблемы технологии и провести их анализ.
2. Произвести анализ доступных технологий и программного обеспечения для реализации проекта.
3. В соответствии с техническим заданием провести разработку мобильного приложения на основе выбранных технологий и программного обеспечения.
4. Подготовить техническую и сопроводительную документацию.

# **Глава 1. Мобильные технологий в образовательном процессе**

## **1.1.Использование мобильных устройств в обучении**

Вот уже не первый год применение мобильных гаджетов в образовательном процессе является актуальной, но в той же мере спорной темой, с одной стороны они позволяют решить многие педагогические проблемы: помощь в демонстрации информации учащимся, замена физических носителей информации, будь то учебники, справочники, энциклопедии, пособия и др., решение примеров задач, проведение тестов и опросов учеников, и вытекающая из этого автоматизация сбора и проверки данных и многое другое, с другой стороны, широко распространено мнение о вреде чрезмерного использования детьми мобильных телефонов. Тем не менее необходимость применения мобильных устройств в образовательной среде возрастает по многим причинам, широкое распространение смартфонов и планшетов у обучаемых и преподавателей, их интерес к применению технологий, которые в значительной степени способны упростить процесс обучения способствуют этому [3].

Стоит оговориться, что понятия “мобильное обучение”, “мобильные технологии” в образовании не одно и то же, что и “мобильное устройство”, которое каждый из нас имеет под рукой. Мобильные образовательные технологии предполагают независимость человека от места и времени получения информации, оно исключает постоянную привязанность к стационарному компьютеру. Мобильное устройство является инструментом для реализации мобильных технологий, это может быть смартфон, ноутбук, планшет, устройство для чтения электронных книг.

Подобные устройства имеют возможность выхода в Интернет, поддерживают множество мультимедийных возможностей и содействуют в решении широкого спектра задач, в частности, относящихся и к педагогике [4].

Одним из преимуществ использования мобильных технологий в ходе процесса обучения является то, что учащиеся получают доступ к контролю

получаемой информации. Свобода выбора среди огромного количества доступных знаний открывает перед студентом неограниченные возможности в обучении, что предоставляет ему свободу и независимость [5]. В отличие от традиционной формы подачи учебных материалов и заданий мобильное обучение применяет инновационные технологии, с помощью которых происходят изменения в процессе преподавания. Для доставки определенного рода информации применяются такие современные сервисы, как социальные сети, облачные хранилища, хостинги медиаданных, а в последнее время некоторые преподаватели стали использовать социальные сети, выкладывая туда полезную информацию [6].

Для удовлетворения образовательных потребностей современного человека становится неизбежным поиск новых и соответствующих нынешнему времени способов донесения информации. Но это станет возможным только тогда, когда все педагоги будут в полной мере владеть и быть активными пользователями информационно-коммуникационных технологий (ИКТ) [7].

Широкий спектр возможностей мобильных приложений позволяет снизить нагрузку на преподавателя, автоматизировать и расширить его возможности в представлении материала, это могут предоставить электронные дневники, сервисы для пользовательских опросов, конструкторы тестов, симуляторы из различных научных областей и многое другое. С точки зрения представления информации: инструкции, видеоуроки, подкасты, скринкасты, обучающие приложения вполне способны существовать автономно без непосредственного участия преподавателя в изучении материала, что предоставляет человеку независимость в выборе места и времени получения информации, а также регулировке затраченного времени на его изучение, в то время как в аудитории/классе отведено определенное количество часов на материал [6].

Исходя из всего вышесказанного можно выделить следующие плюсы использования мобильных технологий в обучении [7]:

- Мобильность.

Смартфоны, планшеты, ноутбуки очень гибкие инструменты для организации учебного процесса, которые не привязаны к месту (в данном случае конкретной аудитории) и времени использования. Возможность комбинирования устройств еще больше расширяет возможности пользователя. Дополнительное использование систем облачного хранения позволяют удаленно хранить необходимую информацию с возможностью ее получения любым устройством, которое имеет доступ к Интернету [8].

- Перманентность образования.

Благодаря мобильности, процесс обучения может проходить постоянно, не ограничиваться аудиторией или работой на стационарном компьютере, временем или днем недели. Все, что необходимо - мобильное устройство с выходом в Интернет.

- Персонализация обучения.

Пользователь в праве самостоятельно выбрать тематику и вид контента в дополнение к основным учебным источникам информации: курсы, лекции, видео, вебинары, приложения, симуляторы, калькуляторы и пр. Оценивая собственные возможности, сам определяет сложность этого материала, имея возможность пропустить основы или наоборот начать с базовых знаний [5].

- Улучшение качеств коммуникации.

Мобильные устройства допускают легкость в выстраивании быстрого и качественного общения между преподавателем и обучающимися, которые могут сами выбрать, какой вид связи удобен для них. Обратная связь с учениками позволяет преподавателям отслеживать показатель успеваемости отдельно по каждому учащемуся.

- Сортировка информации для облегчения поиска по определенным критериям.

Развитие современных поисковых систем в мобильных устройствах очень упростили поиск информации по требованию, достаточно сформировать



необходимый запрос, и поисковик найдет все возможные источники, где встречаются заданные слова.

- Работа с различного рода технологиями.

Работая с большими объемами информации, которые могут предоставляться в разном виде (симуляторы, техническая документация, таблицы, информационные порталы, форумы) учащийся учится извлекать необходимые для его нужд знания и обобщать, видоизменять и применять их.

- Расширение возможностей и обеспечение равного доступа к образованию.

В наши дни мобильные технологии повсеместно используются даже там, где не хватает школ, книг и компьютеров. Цены на мобильные телефоны постоянно снижаются, поэтому все большее количество людей, даже зачастую в бедных регионах, имеют возможность приобрести подобные устройства и знают, как их использовать [4].

Несмотря на упомянутые достоинства выделяют и негативные аспекты использования мобильных устройств в обучающем процессе.

В первую очередь это возможный вред для зрения при длительной работе с устройством. Мнение, сформированное после распространения электронно-лучевых трубок, которые были основой старых мониторов, они могли нанести вред здоровью глаз, но с введением жестких стандартов ТСО начиная с 1992 года о сертификации на эргономичность и безопасность электронного оборудования, данная проблема не является критичной [9]. Принятие стандартов способствовало появлению более безопасных жидкокристаллических панелей, LED- и OLED-мониторов, IPS, LCD.

Несмотря на безопасность современных мониторов, риск ухудшения зрения все же существует. Разрешение экранов ноутбуков и телефонов значительно меньше, чем у стационарного компьютера, что означает меньшую подвижность глазных мышц, это в свою очередь приводит к их ослаблению. Продолжительная работа с небольшими дисплеями приводит к повышенной

нагрузке на глаза, что может привести к близорукости, головным болям, нервному напряжению [10].

Не у всех детей могут быть смартфоны или планшеты. Данная проблема больше характерна для школ, не каждый родитель готов позволить ребенку принести мобильное устройство на уроки, считая его отвлекающим фактором. Возможна ситуация, когда родители не имеют возможности или средств на покупку устройства.

Неготовность учителей менять методику преподавания. Данный недостаток касается более консервативно настроенных специалистов, которые уже привыкли к устоявшимся и проверенным методам обучения. Сложно реорганизовать свою привычную работу, которая практикуется уже много лет.

Контент-фильтрация. При работе с телефонами или планшетами во время учебного процесса у учащегося может возникнуть желание отвлечься на сторонние ресурсы, потому что преподаватель не может следить за работой каждого постоянно. Эта причина часто приводится в качестве довода не использовать мобильные устройства во время уроков, но эта проблема просто решается - работа с мобильными устройствами в учебном заведении подразумевает наличие маршрутизатора, в котором настраивается фильтр, а для отслеживания действий учеников есть специализированные утилиты, которые позволяют отслеживать экраны устройств.

Одновременная зарядка устройств требует тщательной и продуманной подготовки.

С каждым днем мобильные технологии развиваются с большой скоростью, учитываются все прошлые недостатки и ошибки, и разрабатываются новые решения для актуальных проблем. Методики внедрения в образование мобильных устройств все больше укрепляются и становятся нормой в крупных учебных заведениях. Компании, которые выпускают специализированное учебное программное обеспечение, стараются

адаптировать свои продукты для мобильных платформ с учетом разных операционных систем, технических характеристик, разрешений экранов.

В наше время графики используются часто и повсеместно: для домашней работы, для исследований, отчетности, презентаций, личного изучения. Демонстрация поведения графиков функций достаточно полезна как в школе при изучении построения графиков, так и в высших учебных заведениях при исследовании функций. Будучи многофункциональными компьютерами, современные смартфоны могут помочь в этом деле – разумеется, при наличии соответствующих приложений [11].

Несмотря на то, что большое число разработчиков создают различные обучающие приложения лишь малое их количество имеют необходимые качественные характеристики для использования в образовательных учреждениях. Аналогично с приложениями по построению графиков функций, в Play Маркет существует лишь небольшое количество калькуляторов для построения функций с высоким пользовательским рейтингом, но наличие большого количества отзывов, которые хвалят приложения или указывают на их недостатки, показывает заинтересованность пользователей в подобных продуктах.

## **1.2. Выбор инструментов для написания мобильного приложения**

Развитие сенсорных мобильных устройств способствовало развитию специализированных операционных систем, заточенных для работы с сенсорной поверхностью, поэтому преобразилась структура ОС и ее инструменты. У разработчиков появились широкие возможности в разработке действительно отзывчивых сервисов, приложений, платформ, игр, используя новые технологии. В настоящий момент самыми востребованными операционными системами для написания мобильных приложений являются ОС Android (Google Inc.) и iPhone OS/iOS (Apple) [12].

Мобильная операционная система Android - это открытый и бесплатный программный стек Google, который включает в себя операционную систему,

промежуточное программное обеспечение, а также ключевые приложения для использования на мобильных устройствах, включая смартфоны [13].

Существует ряд **достоинств**, которые склоняют выбрать OS Android для написания приложений не только опытных специалистов, но и начинающих разработчиков:

- Низкая стоимость разработки. Написание небольших приложений не требует производительных компьютеров, а все популярные средства разработки, в которых есть все необходимое, распространяются бесплатно. Для размещения своего приложения или игры в Play Маркет нужно единожды заплатить \$25 при регистрации [14].
- Возможность работать с любого компьютера. SDK<sup>1</sup> для разработки приложений Android доступны на Windows, Linux и Mac.
- Разнообразие сред разработки. Возможность выбрать IDE для любых задач, кастомизировать их под необходимые требования.
- Большое комьюнити. Сообщество разработчиков под Android достаточно обширно, много активных форумов, куда можно обратиться за помощью, что является плюсом для начинающих программистов.
- Возможность адаптировать платформу под специализированную задачу. Android является открытым продуктом, что предполагает возможность видоизменять систему под собственные нужды.

Но даже у проверенных и популярных систем могут быть **недостатки**. В случае с OS Android это касается:

- Многообразие девайсов. Для крупных проектов, которые рассчитаны на как можно большее количество пользователей, необходимо учитывать поддержку всех актуальных версий операционной системы,

---

<sup>1</sup> SDK (от англ. software development kit) — набор средств разработки, который позволяет специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

модель устройств и их характеристики, в частности разрешение экранов.

- Актуальность материалов. Android официально существует с 2008 года и постоянно меняется, поэтому разработчику важно отслеживать актуальные материалы и документацию, связанную с его версией системы.

Apple iPhone OS изначально была разработана для использования на устройствах iPhone. Теперь мобильная операционная система называется iOS и поддерживается на ряде устройств Apple, включая iPhone, iPad, iPad 2 и iPod Touch. Мобильная операционная система iOS доступна только на устройствах Apple собственного производства, поскольку компания не лицензирует ОС для стороннего оборудования. Apple iOS является производной от операционной системы Apple Mac OS X.

Разработчики приложений под iOS выделяют следующие достоинства работы с платформой:

- Определенность. Разработчики всегда знают, какие версии устройств поддерживаются системой, характеристики всегда определены и постоянны.
- Востребованность. Количество разработчиков, работающих с iOS, значительно меньше, чем специалистов Android, поэтому на них всегда есть спрос на рынке.
- Строгая модерация App Store. Для того, чтобы приложение или игру можно было разместить в магазине, необходимо пройти проверку, которая может длиться неделями, прежде чем модерация сообщит свое решение. По итогу, в AppStore приложения в большинстве своем безопасны и имеют высокое качество [15].

Но несмотря на положительные стороны разработки под iOS, существуют значительные сложности для рядового разработчика в виде:

- Большие начальные затраты. Для начала разработки программисту необходимы стартовые вложения в виде покупки iPhone и Mac (компьютер или ноутбук), что требует очень больших денежных затрат. Ко всему прочему, для возможности публикации продуктов в магазине приложений необходимо каждый год выплачивать сумму в \$99.
- Длительность ожидания публикации приложения. Каждое приложение при загрузке в AppStore проходит тщательную проверку модераторами магазина, весь процесс может растянуться на месяц, после которого, если продукт не удовлетворяет установленным критериям, разработчику могут отказать в публикации [15].

Таблица 1.

Сравнительная таблица операционных систем iOS и Android

	iOS	Android
Начальные вложения.	Высокие	Низкие
Аппаратное и программное обеспечение.	Возможность разработки только на устройствах Apple	Возможность разработки на любой доступной ОС: Windows, Linux, Mac.
Многообразие девайсов и их поддержка .	iPhone, iPad и iPod Touch, умные девайсы; Каждый поддерживаемый девайс определен, у каждой модели постоянные характеристики.	Огромное количество устройств: смартфонов, планшетов, умных девайсов разных производителей, характеристики которых могут различаться от модели, серии, производителя. Нельзя точно сказать, когда версия системы устареет.
Доступ к исходному коду.	Нет.	Да.
Разнообразие IDE.	Многие IDE имеют возможность работать и с iOS и с Android	
Комьюнити.	Большое.	Большое.
Количество обучающего	Большое.	Большое, но из-за разнообразия версий

материала.		системы, многие могут быть не актуальными.
Стоимость публикации продукта в магазине приложений.	\$99 в год (7004,45Р на май 2020 г.).	Один раз при создании аккаунта разработчика в размере \$25 (1768,8Р на май 2020 г.).
Модерация магазина приложений.	Строгая, проверка может длиться недели.	Автоматическая, ручная проверка происходит только после большого количества жалоб на приложение.
Востребованность.	Количество разработчиков, чья специальность iOS, значительно меньше, поэтому такие специалисты востребованы.	Количество разработчиков под Android много из-за низкого порога вхождения, поэтому спрос на подобных специалистов не очень высокий.

Независимые разработчики или просто новички часто сталкиваются с проблемой выбора операционной системы для написания своего продукта, потому что у них нет заказа на приложение с уже указанной ОС, этот вопрос им приходится решать самостоятельно. Статистика показывает, что нет особой зависимости выбора операционной системы от достатка человека, существует мнение, что продукцию Apple могут позволить себе далеко не каждый человек, но как показывают карты распределения смартфонов по некоторым городам России, выбор больше зависит от региона проживания человека [16]. iOS чаще встречается среди жителей Москвы и Санкт-Петербурга, в остальных регионах преобладает OS Android, но точного распределения по регионам нет.

Решение данного вопроса больше зависит от того, для кого разрабатывается продукт, кем будут его основные пользователи. Немалое значение отводится предпочтению и уже имеющимся ресурсам самого разработчика, с какими инструментами он имеет опыт работы (см. *Таблица 1*).

Для разработки мобильного приложения для построения графиков функций было решено использовать операционную систему Android, она требует минимальных начальных ресурсов и имеет много разнообразных инструментов для реализации задачи. Данная OS имеет широкое распространение среди учеников школ и высших учебных заведений, именно на них ориентирован продукт, поэтому выбранная операционная система нам подходит.

Следующим необходимым шагом будет выбор модели разрабатываемого приложения. Важный и необходимый шаг в разработке приложения, который определяет техническую оценку, ориентированность на платформу, доступ к информации на устройстве, необходимость в доступе к Интернету. Они подразделяются на 4 вида [17]:

1. Нативные.
2. Веб-приложения.
3. Гибридные.
4. Кроссплатформенные.

Нативные приложения создаются для одной определенной платформы, с последующей установкой на устройство пользователя. Именно подобные продукты разработки загружаются в магазины приложений. Часто нативные приложения создаются компаниями в качестве мобильной альтернативы основного веб-сервиса.

Подобные программные продукты имеют возможность собирать доступную информацию о пользователе из его смартфона, например, местоположение, время работы в приложении, данные связи, характеристики телефона, логи<sup>2</sup> и/или коды ошибок при экстренном прекращении работы программы и др., это не обязательно что-то негативное, чаще всего данная информация используется для исправления существующих ошибок в

---

<sup>2</sup> Файл регистрации (протокол, журнал, лог; англ. log) — файл с записями о событиях в хронологическом порядке, простейшее средство обеспечения журналирования.



приложении, навигации, статистики качества связи, персонализации рекламы и многое другое.

Веб-приложения работают через браузер смартфона и являются оптимизированными сайтами для отображения на мобильном устройстве. Имеют вид нативных приложений, но не устанавливаются на девайс пользователя, при качественном написании приложения может не уступать нативному. Веб-приложения функционируют на любой операционной системе, и их легко разрабатывать, достаточно знать HTML, CSS и JS.

Веб-приложения не имеют доступа к устройству пользователя, и для их использования необходимо подключение к Интернету.

Гибридные приложения, как следует из названия, совмещает в себе элементы нативных и веб-приложений. Подобные продукты выглядят и используются, как нативные, они размещаются в магазинах приложений и устанавливаются на смартфон пользователя, но работают через WebView<sup>3</sup>.

Данные приложения обладают большой функциональностью, не ограничены определенной операционной системой, но разработка сложнее по сравнению с веб-приложениями [17].

Завершает список кроссплатформенная модель приложений. Они очень похожи на гибридные, потому что со своей сути не привязаны к определенной операционной системе, но методы достижения этого у них разные. Кроссплатформенная разработка нацелена на написание большей части кода приложения не зависящей от дальнейшего выбора платформы пользователем.

Следующим шагом в выборе инструментов для реализации приложения будет выбор языка программирования. Можно опираться на выбор языка по предпочитаемой модели разрабатываемого продукта, а можно по уже изученному или заранее определенному языку и параметрам разработки решить каким именно будет вид приложения.

---

<sup>3</sup> WebView — это компонент платформы Android, который позволяет встраивать web-страницы в Android-приложения, по сути, это встраиваемый браузер.

Выбор языка программирования может быть непростой задачей, для разработчика важно использовать актуальные на момент написания программы инструменты, от этого может зависеть итоговая совместимость с устройством, техническая поддержка, актуальность материалов, наличие активного сообщества и многое другое. На начало 2020 г. определились следующие популярные языки программирования для разработки под iOS и Android [18]:

### **Swift**

Язык, созданный компанией Apple, предназначен для разработки приложений для iOS, OS X, iPadOS, MacOS и WatchOS. Язык много позаимствовал из C++ и Objective-C. Простой и функциональный язык, легок в изучении, но по словам специалистов, имеет плохую документацию Apple [19].

Популярная среда разработки: Xcode.

### **Kotlin**

Статически типизированный язык программирования, разработанный компанией JetBrains, рассчитан на разработку под OS Android. Полностью совместим с Java, что упрощает переход с одного языка на другой. Функциональный язык, который имеет много обучающих курсов и материалов от разработчиков и Google, но имеет небольшое, но развивающееся сообщество [20].

Популярная среда разработки: Android Studio.

### **Dart**

Язык программирования Google, альтернатива JavaScript. Используется в комбинации с Flutter, что позволяет создавать приложения для всех основных платформ - Android, Windows, macOS, Linux, iOS, Web. Является кроссплатформенным, активно развивается и поддерживается, но, как и у предыдущего языка небольшое сообщество, нестабилен [21].

Популярная среда разработки: Android Studio.

### **JavaScript**

Прототипно-ориентированный сценарный язык программирования, широко применяется в веб-разработке, также активно используется в различных фреймворках. Платформы для разработки: iOS, Android, Web. Легок для изучения, имеет огромное сообщество, но при использовании его в написании приложения может пострадать скорость работы, обязателен фреймворк [22].

Популярные среды разработки: Atom, Android Studio, Xcode, блокнот Notepad++ и другие.

### **Java**

Строго типизированный объектно-ориентированный язык программирования. Является самым востребованным языком на рынке, но в нише разработки для мобильных устройств уступает современным специализированным языкам. Платформа для разработки: Android. Имеет большую популярность и сообщество, но имеет сложный код и труден в изучении [23].

Популярная среда разработки: Android Studio.

Существует еще много языков программирования, которые используются в мобильной разработке, но мы рассмотрели лишь самые популярные в настоящий момент.

Ранее был упомянут вспомогательный инструмент разработки - фреймворк. Под понятием фреймворк понимается специальная программная среда выполнения, программный каркас, который облегчает написание приложений для разработчика и объединяет разные компоненты большого программного проекта, содержит в себе некоторую не меняющуюся основу - структуру, а пользователю необходимо заполнить эту основу необходимыми ему модулями.

Фреймворки очень популярный инструмент разработки мобильных приложений, они способны из кода на HTML, CSS и JS (либо ему подобные) создать полноценное нативное или гибридное приложение, нацеленное на

увеличение производительности, сокращение времени и усилий на разработку. За время развития инструментов для разработки мобильных приложений появилось огромное множество фреймворков и библиотек, которые облегчают работу программиста, поэтому рассмотрим лишь некоторые из них, которые будут полезны для нашей конкретной задачи [24].

### **Angular**

Фреймворк от компании Google для создания клиентских приложений, удобен для создания одностраничных приложений. Использует в качестве рабочих языков программирования TypeScript, Dart, JavaScript, но TypeScript является основным [25].

### **React Native**

Мобильный фреймворк созданный на базе React JS. Позволяет разрабатывать кроссплатформенные приложения. Использует JavaScript и React.js [26].

### **PhoneGap**

Бесплатный open-source фреймворк, использует чистый JS в совокупности с HTML и CSS. После написания код “обертывается” Apache Cordova и запускается, как кроссплатформенное приложение [27].

### **Ionic**

Open-source фреймворк для создания гибридных приложений с использованием HTML, CSS и JavaScript. Построен на стеке Cordova и работает совместно с Angular [28].

Завершающей задачей для программиста является выбор среды разработки (IDE), их количество невелико и каждый из них поддерживает определенное количество языков, поэтому выбор будет зависеть от требуемого языка и поставленной цели.

### **Microsoft Visual Studio**

Функционал IDE позволяет создавать множество типов программ: от веб-приложений до игр. Является очень гибкой средой разработки, все

необходимые пакеты инструментов можно скачать как во время основной загрузки продукта, так и после установки. Имеет возможность анализа кода, отладки, тестирования, развертывания [29]. Предоставляет возможность работы с фреймворками.

Поддерживаемые языки: Ajax, ASP.NET, DHTML, JavaScript, JScript, Visual Basic, Visual C#, Visual C++, Visual F#, XAML и другие.

### **PyCharm**

IDE разработанная командой Jet Brains, имеет бесплатную немного урезанную версию Community Edition, основной поддерживаемый язык - Python, но способен поддерживать и некоторые другие: AngularJS, Coffee Script, CSS, Cython, HTML, JavaScript, Node.js, TypeScript [30].

### **Eclipse**

Бесплатный и гибкий редактор с открытым исходным кодом. Изначально создавалась как среда для написания Java-программ, но постепенно расширилась до поддержки C, C++, Java, Perl, PHP, Python, Ruby и других. Имеет широкий диапазон возможностей благодаря большому количеству плагинов, инструменту отладки, поддержке Git, Plugin Development Tooling [31].

### **Android Studio**

Специализированная IDE от компании Google разработана на основе IntelliJ IDEA, заточена на разработку под Android. Имеет возможность наблюдения за изменениями в проекте в режиме реального времени [32].

Поддерживаемые языки: Java, C++, Kotlin.

Рассматривая все выбранные инструменты, нужно определиться, какие средства разработки будут более подходящими для наших целей. Главным условием было использовать элемент HTML5 Canvas для вывода изображения построенного графика функции, поэтому предпочтительным будет использовать JavaScript в качестве основного языка, а фреймворк позволит преобразовать полученный код в приложение для мобильного устройства.

Фреймворки для разработки мобильных приложений хоть и разнообразны, но принцип работы един, поэтому выбираем Ionic, который имеет возможность создавать уже готовые шаблоны одностраничных приложений. Для корректной работы программы не нужен доступ в Интернет, она должно работать, как самостоятельный инструмент, поэтому вид приложения выбираем нативный. В качестве среды разработки выберем Android Studio, в котором есть необходимые нам SDK, эмулятор Android для тестирования приложения.

### **1.3.Формализованное описание технического задания**

#### **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

на разработку информационной системы

Мобильное приложение для построения простейших графиков функций с использованием Canvas

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы».

#### **1. Общие сведения.**

##### **1.1.Название организации-заказчика.**

Федеральное государственное бюджетное образовательное учреждение высшего образования «Уральский государственный педагогический университет»

##### **1.2.Название продукта разработки (проектирования).**

Мобильное приложение для построения простейших графиков функций с использованием Canvas.

##### **1.3.Назначение продукта.**

Мобильное приложение разработано с целью упрощения и автоматизации построения графиков функций для людей, кому в рамках задания или личных нужд необходимо без лишних затрат построить графики функций.

##### **1.4.Плановые сроки начала и окончания работ.**

В соответствии с планом выполнения ВКР (с 01.09.2019 по 19.05.2020).

#### **2. Характеристика области применения продукта.**

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Продукт может быть использован в качестве вспомогательного инструмента для построения графиков функций в школах, колледжах, институтах, для выполнения личных или производственных расчетов.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Для работы с приложением пользователю необходимо иметь базовые знания работы со смартфоном, знать правила записи функций.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

Приложение не должно быть требовательным к аппаратному и системному обеспечению, должно быть интуитивно понятным для пользователя и иметь большой выбор функций для построения.

3.2. Аппаратные требования.

Для нормального функционирования приложения нет определенных аппаратных требований, пользователю желательно иметь смартфон с 1Гб оперативной памяти, частотой 4\*1.3 ГГц.

3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).

Для нормального функционирования приложения необходимы следующие системные требования - ОС Android версии 5.0 и выше, наличие установленного Google Play Маркет.

3.4. Указание программного обеспечения, используемого для реализации.

Microsoft Visual Studio, Ionic Framework.

3.5. Форматы входных и выходных данных

Входные данные: число(a), формула.

Выходные данные: изображение графика функции.

3.6. Источники данных и порядок их ввода в систему (программу), порядок вывода, хранения.

В качестве источника данных выступает сам пользователь, который вводит необходимые числовые данные в заданные поля, программа обрабатывает их вместе с выбранной формулой и выводит на экран смартфона построенный график.

3.7. Порядок взаимодействия с другими системами, возможности обмена информацией.

Не предусмотрено.

3.8. Меры защиты информации.

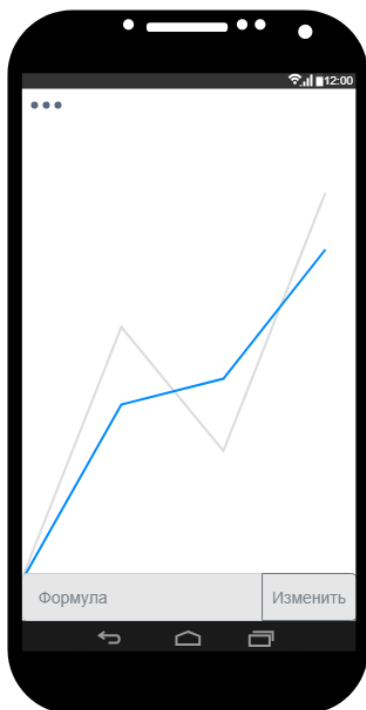
Не предусмотрено.

4. Требования к пользовательскому интерфейсу.

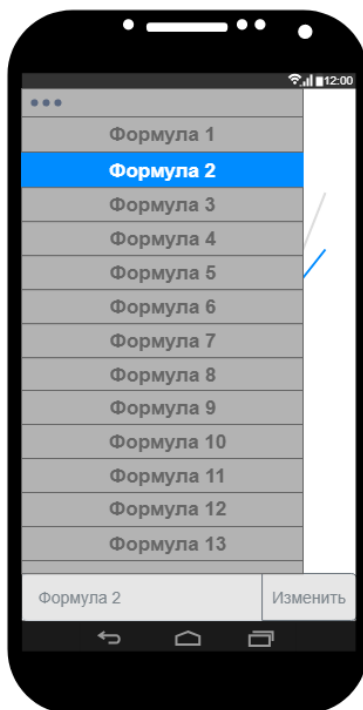
4.1. Общая характеристика пользовательского интерфейса.

Минималистичный пользовательский интерфейс, интуитивно понятный для широкого круга пользователей.

4.2. Размещение информации на экране, дизайн экрана.



а)



б)



в)



Рис 1. Дизайн интерфейса приложения: а) главный экран; б) меню выбора формулы;  
в) пример ввода данных пользователем в формулу.

#### 4.3. Особенности ввода информации пользователем, представление выходных данных.

Пользователь вводит числовые данные необходимые для выбранной функции, программа считывает данные и на их основе строит график функции, который выводится на экран.

#### 5. Требования к документированию.

##### 5.1. Перечень сопроводительной документации.

Руководство пользователя.

#### 6. Порядок сдачи-приемки продукта.

В соответствии с планом выполнения ВКР.

## **Глава 2. Разработка мобильного приложения**

### **2.1. Модельное представление мобильного приложения**

Глобальной задачей разработки является написание приложения для построения графиков функций, главное условие которого использовать элемент Canvas. Мобильное приложение должно быть ориентировано на построение простых функций, из расчета, что пользоваться программой будут школьники и студенты колледжей/вузов.

Макет приложений для построения уравнений и/или функций в магазине приложений Play Маркет можно разделить на математические калькуляторы, где ввод функции производится из панели с математическими функциями, коэффициентами, константами и операторами, или на одностраничные минималистичные приложения, где пользователь самостоятельно вводит формулу по заданным правилам.

Вид калькулятора хоть и дает гибкость построения различных функций, но для демонстрации поведения графиков и обучающего материала не подходит, т.к. имеет намного больший функционал, чем это необходимо.

Вид простого одностраничного приложения с меню, где будет находиться список функций, будет более удачным для обучающей программы (см. Рис 1).

Перед разработкой программного продукта целесообразно составить такие диаграммы, как IDEF0 и UML, которые в значительной степени облегчают написание программного продукта, графически описывают структуру проекта, взаимосвязь между функциональными элементами, позволяют взглянуть на систему в определенный момент ее выполнения. Подобные методы описания являются очень хорошей презентацией внутренней структуры продукта.

IDEF0 – это графический стандарт для функционального моделирования, который предназначен для формализации и описания процессов системы. В нем рассматриваются логические отношения между процессами, а не временная последовательность. Имеет вид «черного ящика» с входами, выходами,

управлением и механизмом (см. Рис 2), который по необходимости может иметь несколько уровней детализации (см. Рис 3).

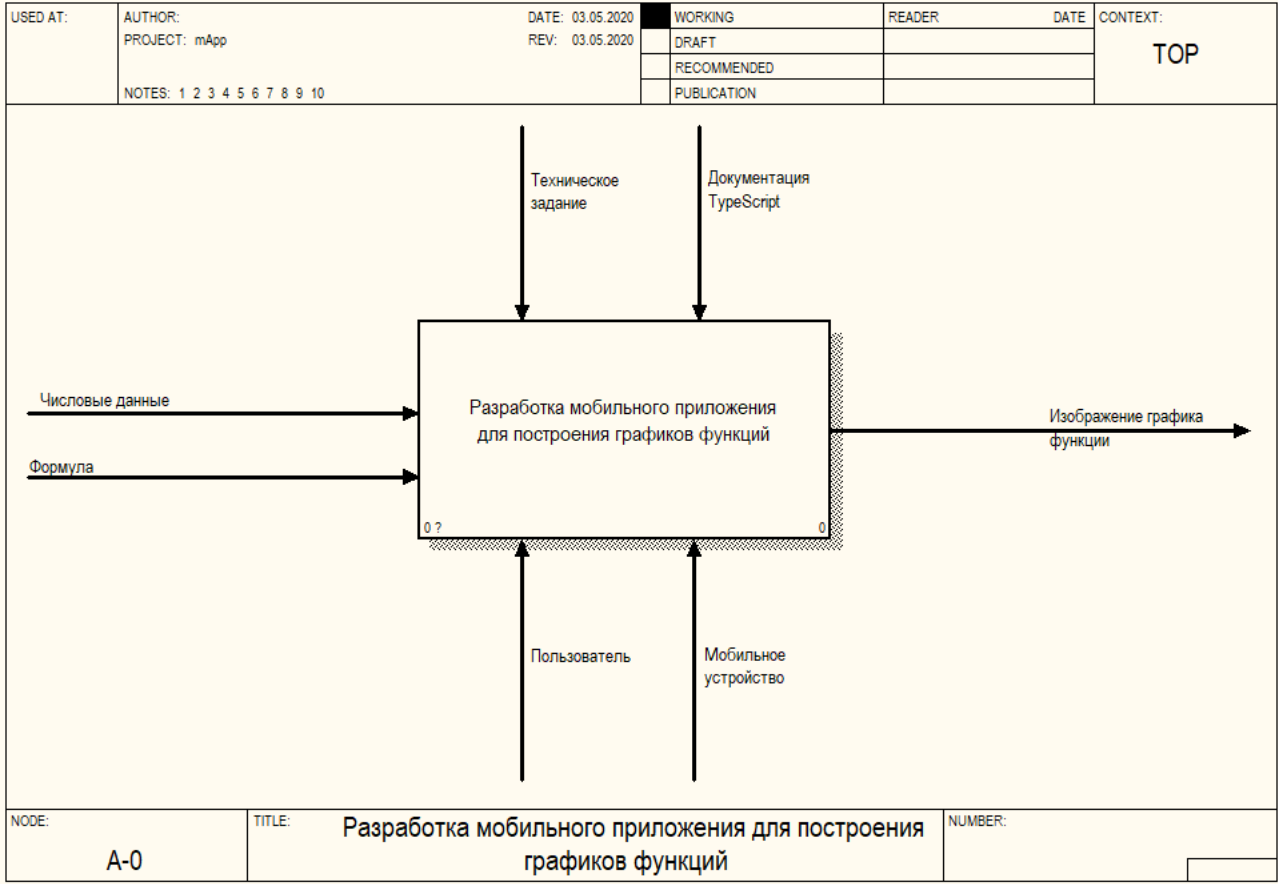


Рис 2. Диаграмма IDEF0 разработки мобильного приложения

На вход приложение получает выбранную пользователем формулу и числовые данные, которые являются коэффициентами формулы, это те вводные, которые необходимы для начала работы. На выходе приложение создает изображение графика выбранной функции. Управляющими факторами для разработки приложения являются техническое задание и документация TypeScript (далее TS). Механизмами являются пользователь и его мобильное устройство. Используя смартфон, пользователь будет выбирать нужную формулу и вводить ее коэффициенты, таким образом выступая в качестве источника данных. Таким образом задаются основные параметры процесса, учитывается все необходимое для успешной реализации поставленной задачи.

Следующий уровень детализации имеет 4 процесса.

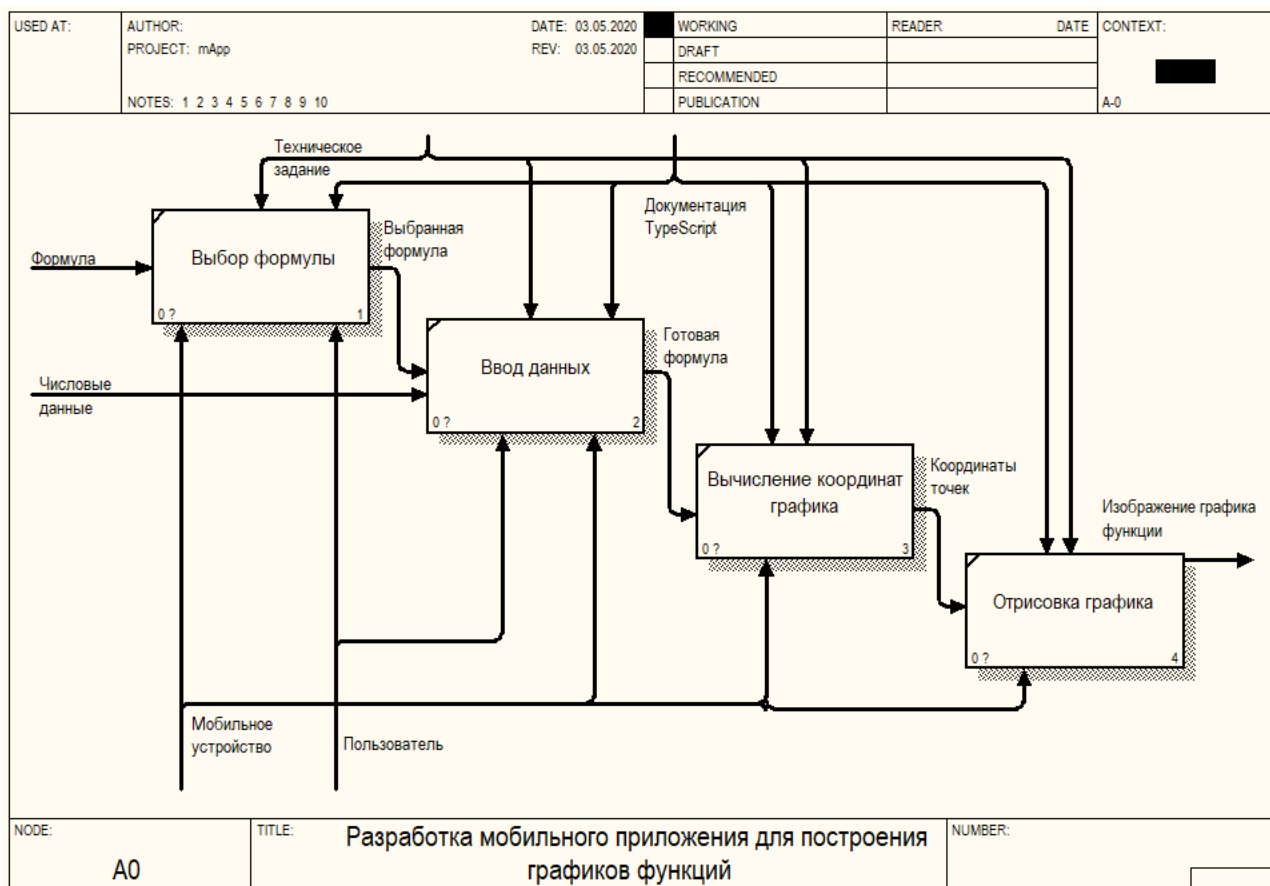


Рис 3. Диаграмма декомпозиции 1 уровня

На схеме можно наглядно увидеть, на какие процессы влияет сам пользователь, в какие процессы передаются входные данные.

Выбор формулы предполагает участие пользователя и мобильного устройства на котором происходит процесс, на вход пользователем передается выбранная формула.

Ввод данных также требует участия мобильного устройства и пользователя, который должен задать числовые коэффициенты в формулу.

Вычисление координат графика принимает на вход сформированную формулу, которая с помощью инструментов TS выдает набор координат, которые подаются на вход процессу по отрисовке графика. В этих процессах пользователь не принимает участие, а всю необходимую работу выполняет приложение. В качестве результата написанная программа выводит

изображение построенного графика. Выполнены процессы в соответствии с техническим заданием и документацией TS.

IDEF0 диаграмма рассматривает задачу с точки зрения основных процессов необходимых для ее реализации, учитывая входные данные, результат, который мы получаем на выходе, влияющие на разработку механизмы и управление.

UML диаграммы применяются для объектно-ориентированного анализа, визуализации, проектирования и документирования программных систем. Они предоставляют возможность посмотреть на задачу с разных точек зрения, помогают легче понять суть задачи. Диаграммы очень просты для чтения. UML является стандартом, который использует графические обозначения для реализации абстрактной модели создаваемой системы.

UML диаграммы подразделяются на диаграммы классов, компонентов, составной структуры, объектов, пакетов, деятельности, автомата, вариации использования, коммуникации и последовательности и некоторые другие.

Исходная задача по написанию мобильного приложения не является сложной для восприятия или очень обширной, поэтому не требует большого количества UML диаграмм для более детального рассмотрения каждого элемента и этапа разработки. В дополнение к IDEF0 была составлена диаграмма последовательности (см. Рис 4), которая рассматривает взаимодействие объектов системы во времени.

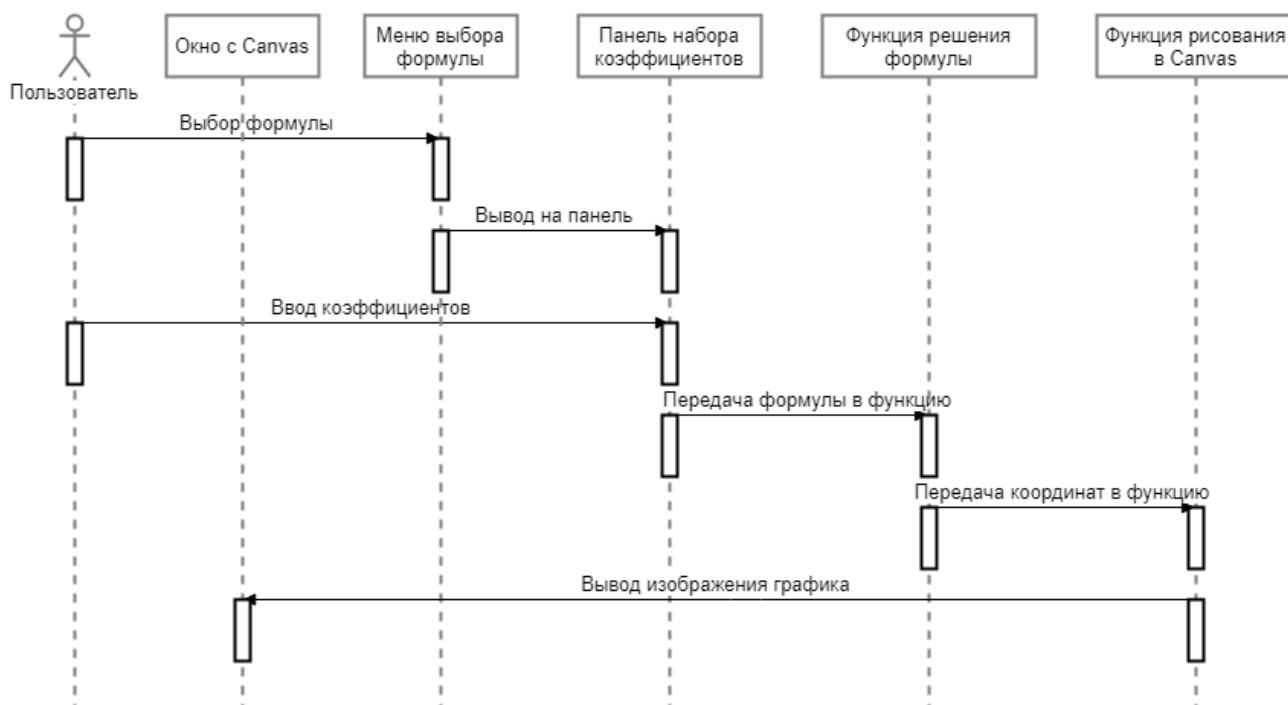


Рис 4. Диаграмма последовательности для разрабатываемого приложения

По данной диаграмме мы видим, что основным действующим лицом является пользователь, который посредством работы с интерфейсом приложения в первую очередь должен выбирать формулу, а затем ввести нужные коэффициенты. Заполненная формула передается в функцию, чья задача решить уравнение на определенном интервале. Полученные значения передаются в функцию, которая, используя инструменты TS, нарисует график по данным, они в свою очередь отобразятся на странице приложения, где размещен Canvas.

Выбрав фреймворк в качестве основы для написания приложения, задается определенная структура программы. Несмотря на то, что все фреймворки могут сильно отличаться друг от друга, есть определенные вещи универсальные для любых проектов. MVC — Модель, Представление и Контроллер (Model-View-Controller) — три составляющих каждого веб-фреймворка (см. Рис 5) [33].



Рис 5. Схема MVC

Модель содержит все введенные данные, правила и функции.

Представление отвечает за визуальное отображение содержимого - это интерфейс, стили, Canvas.

Контроллер управляет данными предыдущих двух элементов.

Для программиста подобный шаблон очень упрощает процесс разработки, и экономит время для написания основного функционала программы. Проблема может появиться, когда проект становится большим, возрастает нагрузка на контроллер, и приложение, написанное на основе фреймворка, может потерять в быстродействии. Разрабатываемая программа не нацелена на обширный функционал калькулятора, поэтому скорость работы должна быть оптимальной. Согласно книге «MEAN<sup>4</sup> Web Development» от Amos Naviv для небольших проектов очень полезна горизонтальная структура папок, где количество функций ограничено, поэтому файлы удобно размещать внутри папок, которые представляют их общие роли.

## 2.2. Описание приложения

В процессе разработки было написано небольшое учебное приложение, основной функцией которого является построение простейших графиков функций с применением технологии HTML5 Canvas. С помощью фреймворка

---

<sup>4</sup> MEAN (аббревиатура от MongoDB, Express.js, Angular.js, Node.js) — набор серверного программного обеспечения, который используется для веб-разработки, ориентированный на JavaScript: все компоненты стека поддерживают программирование на JavaScript, и серверная и клиентская часть MEAN-приложений может быть написана на этом языке программирования.

Ionic, который позволяет создавать готовый шаблон главной страницы (см. Рис 6), из-за чего мы могли сосредоточиться на написании функционального содержимого программы. Для написания приложения наш выбор остановился на шаблоне `sidemenu`, который предоставляет наличие всплывающего меню в левой части окна приложения.



Рис 6. Варианты выбора шаблона приложения, предлагаемые IonicFramework

Для разработки проекта понадобился ряд других программ, только после их настройки можно было использовать функционал фреймворка. В этот список входит установка и настройка:

- Node.js – это среда выполнения, которая позволяет писать на JavaScript на стороне сервера. Помимо использования для веб-служб, Node часто используется для создания инструментов разработчика, таких как Ionic CLI<sup>5</sup>. Почти все инструменты для современных проектов JavaScript основаны на Node.js.
- npm - это менеджер пакетов для Node, он позволяет разработчикам устанавливать, совместно использовать и упаковывать модули Node.
- Git – это распределенная система управления версиями, выражаясь проще, это программа записывающая изменения проекта в течение определённого времени, что позволяет при необходимости вернуться к старой версии проекта. Согласно документации, установка Git не

---

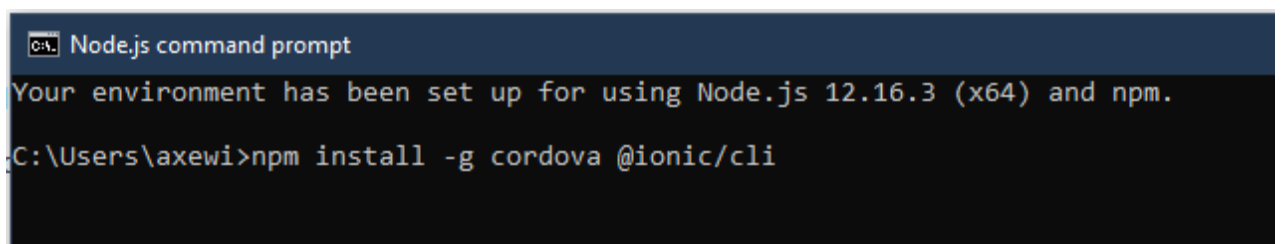
<sup>5</sup> CLI, или интерфейс командной строки, представляет собой текстовый интерфейс для взаимодействия с программой. Ionic CLI можно использовать для различных целей, таких как создание проекта и его сборка в приложение, запуск сервера.



обязательна, но настоятельно рекомендуется. Аналогично проверим корректность Git.

- Apache Cordova - это платформа разработки мобильных приложений с открытым исходным кодом, которая позволяет преобразовать стандартный HTML/CSS/JS код в полноценное нативное или кроссплатформенное приложение. Предоставляет возможность доступа к датчикам устройства, данным и состоянию сети [34].
- Angular, React, или Vue. Многие фреймворки JS не являются самодостаточными, предоставляя свой шаблон и синтаксический сахар<sup>6</sup> для JS или TS, поэтому часто для написания приложения они требуют интеграцию с другими самостоятельными фреймворками.
- Android Studio. Удобная и функциональная среда разработки, которую рекомендует установить фреймворк ради встроенных Android SDK, эмулятора Android, для теста приложения на уже готовой операционной системе, а не в браузере.

Только после настройки утилит, которая происходит в основном через командную строку и `npm` (см. Рис 7), появляется возможность создать свой проект (см. Рис 8). Как уже было отмечено ранее, программы, основанные на фреймворках, имеют определенную структуру папок, которые объединены по выполняемым функциям (см. Рис 9).



```
Node.js command prompt
Your environment has been set up for using Node.js 12.16.3 (x64) and npm.
C:\Users\axewi>npm install -g cordova @ionic/cli
```

Рис 7. Скриншот команды установки пакета Ionic вместе с Cordova

---

<sup>6</sup> Синтаксический сахар (англ. syntactic sugar) в языке программирования — это синтаксические возможности, применение которых не влияет на поведение программы, но делает использование языка более удобным для человека.

```
Командная строка
Microsoft Windows [Version 10.0.18363.720]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.
C:\Users\axewi>ionic start graph sidemenu
```

Рис 8. Команда создания проекта по шаблону sidemenu

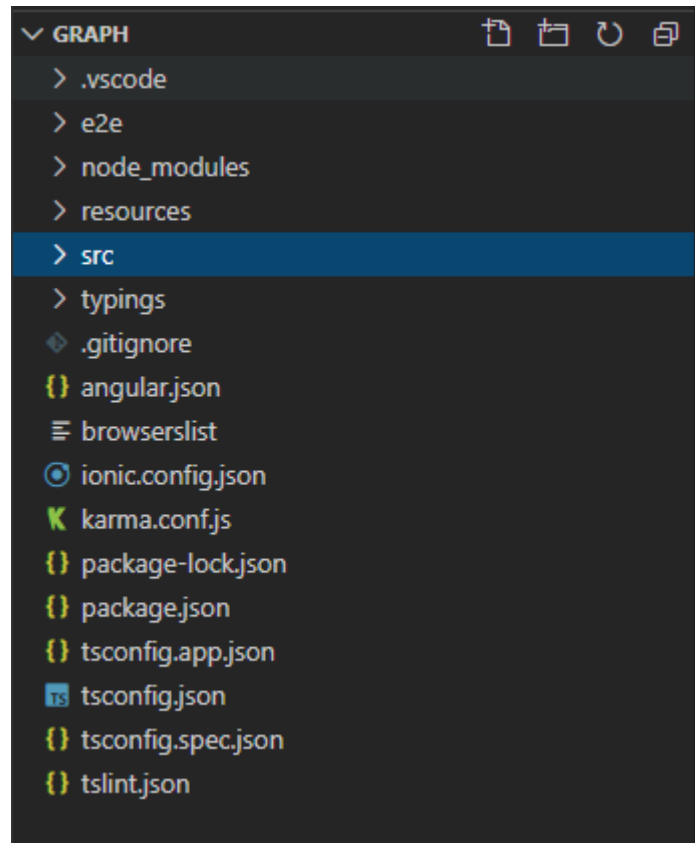


Рис 9. Структура папок проекта на IonicFramework

У Ionic для шаблона структура документа состоит из 6 папок, но для нас имеют значение лишь папки resources, src и файлы json, которые находятся все папок. Папка ресурсов содержит картинки и иконки, которые мы используем в приложении, папка src представляет для нас наибольший интерес, т.к. именно в ней находятся файлы html, css и ts, в которых и написан весь код программы(см. Рис 10).

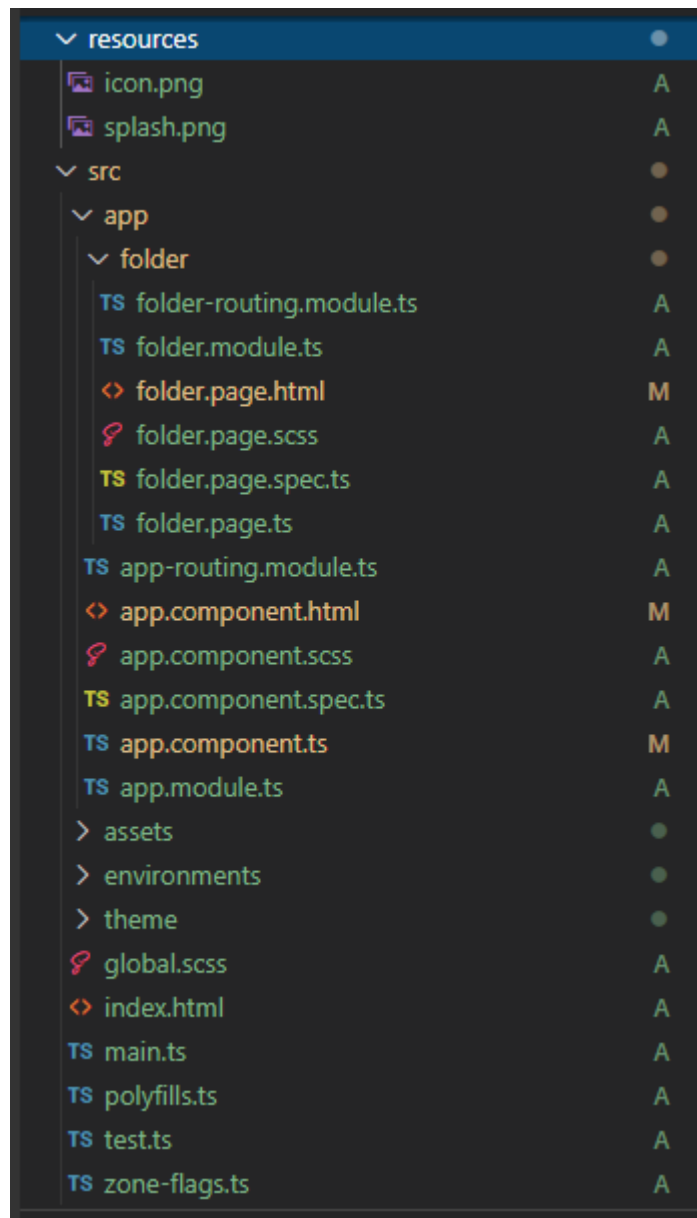


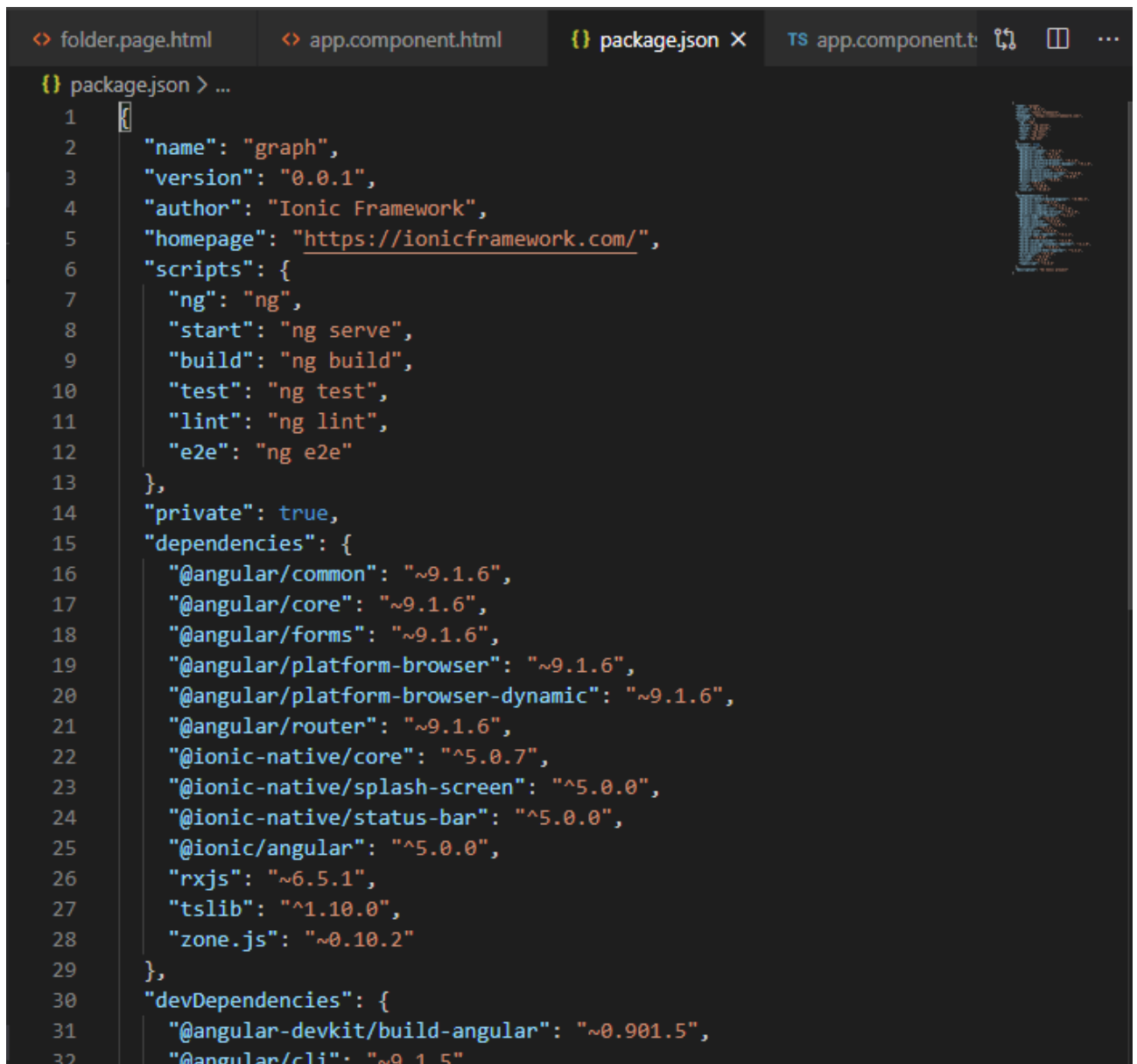
Рис 10. Содержимое папок resources и src

Папка themes содержит предустановленные варианты тем приложения, прописанные цвета для корректного отображения на разных платформах, вариации цветов подсказок, кнопок, панелей и многого другого (см. Рис 11).

```
src > theme > variables.scss > ...
1  |// Ionic Variables and Theming. For more info, please see:
2  |// http://ionicframework.com/docs/theming/
3
4  /** Ionic CSS Variables **/
5  :root {
6    /** primary **/
7    --ion-color-primary: #5260ff;
8    --ion-color-primary-rgb: 82, 96, 255;
9    --ion-color-primary-contrast: #ffffff;
10   --ion-color-primary-contrast-rgb: 255, 255, 255;
11   --ion-color-primary-shade: #4854e0;
12   --ion-color-primary-tint: #6370ff;
13
14   /** secondary **/
15   --ion-color-secondary: #3dc2ff;
16   --ion-color-secondary-rgb: 61, 194, 255;
17   --ion-color-secondary-contrast: #ffffff;
18   --ion-color-secondary-contrast-rgb: 255, 255, 255;
19   --ion-color-secondary-shade: #36abe0;
20   --ion-color-secondary-tint: #50c8ff;
21
22   /** tertiary **/
23   --ion-color-tertiary: #5260ff;
24   --ion-color-tertiary-rgb: 82, 96, 255;
25   --ion-color-tertiary-contrast: #ffffff;
26   --ion-color-tertiary-contrast-rgb: 255, 255, 255;
```

Рис 11. Скриншот файла с темами оформления

В файлах json указывается конфигурация проекта, подключаемые библиотеки и ядра, версии программ, которые мы используем, команды для запуска, сборки, тестирования проекта (см. Рис 12).



```
1  {} package.json > ...
2  {
3    "name": "graph",
4    "version": "0.0.1",
5    "author": "Ionic Framework",
6    "homepage": "https://ionicframework.com/",
7    "scripts": {
8      "ng": "ng",
9      "start": "ng serve",
10     "build": "ng build",
11     "test": "ng test",
12     "lint": "ng lint",
13     "e2e": "ng e2e"
14   },
15   "private": true,
16   "dependencies": {
17     "@angular/common": "~9.1.6",
18     "@angular/core": "~9.1.6",
19     "@angular/forms": "~9.1.6",
20     "@angular/platform-browser": "~9.1.6",
21     "@angular/platform-browser-dynamic": "~9.1.6",
22     "@angular/router": "~9.1.6",
23     "@ionic-native/core": "^5.0.7",
24     "@ionic-native/splash-screen": "^5.0.0",
25     "@ionic-native/status-bar": "^5.0.0",
26     "@ionic/angular": "^5.0.0",
27     "rxjs": "~6.5.1",
28     "tslib": "^1.10.0",
29     "zone.js": "~0.10.2"
30   },
31   "devDependencies": {
32     "@angular-devkit/build-angular": "~0.901.5",
33     "@angular/cli": "~9.1.5",
```

Рис 12. Скриншот файла package.json

Файлы проекта, в которых мы писали код для главной страницы с Canvas находятся в папке folder и имеют названия folder.page.html, folder.page.scss и folder.page.ts. Всплывающее меню прописано в папке app в файлах app.component.html и app.component.ts (см. Рис 13).



```
<ion-menu-toggle auto-hide="false" *ngFor="let p of appPages; let i = index">
  <ion-item (click)="selectedIndex = i" routerDirection="root" [routerLink]="[p.url]" lines="none" detail="false" [class.selected]="selectedIndex == i">
    <ion-icon slot="start" [ios]="p.icon + '-outline'" [md]="p.icon + '-sharp'"></ion-icon>
    <ion-label>{{ p.title }}</ion-label>
  </ion-item>
</ion-menu-toggle>
</ion-list>
```

Рис 13. Пример вывода в меню списка формул из файла app.component.html

По итогу разработки мы получили минималистичное, интуитивно понятное приложение, с изображением графика на главной странице (см. Рис 14), благодаря элементу Canvas, и возможности самому выбирать количество нужных координат с помощью TS, мы получаем плавный и четкий график. Меню с выбором функции (см. Рис 15) работает плавно, приложение при работе на эмуляторе работало без задержек.

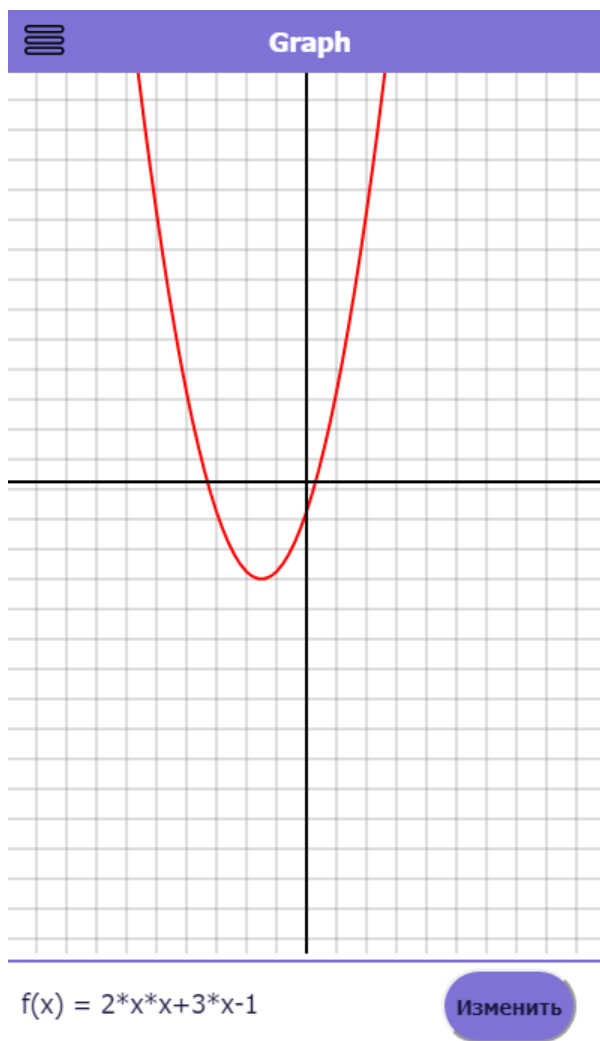


Рис 14. Скриншот главного экрана приложения

По нажатию на строку с выбранной формулой, у пользователя есть возможность изменить числовые коэффициенты в функции, подтвердить свои изменения можно кнопкой «Изменить», после чего измененная формула считывается скриптом, передаётся функции расчета (см. ), где вычисляются

координаты графика. Они передаются в функцию отрисовки, которая связана с Canvas на главной странице.

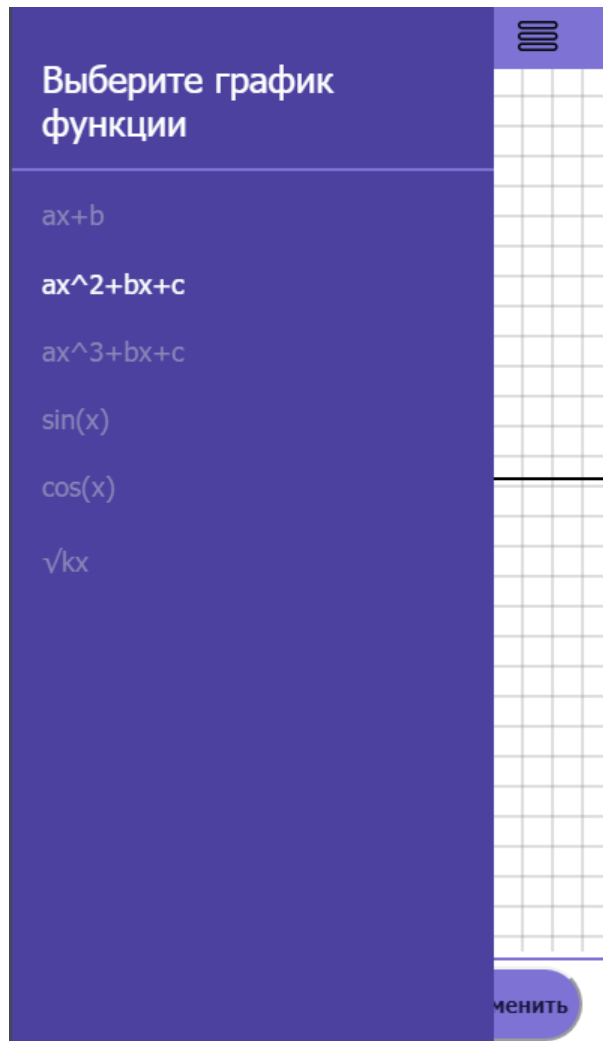


Рис 15. Скриншот меню приложения

## 2.3. Сопроводительная документация

### Руководство пользователя

#### 1. Требования к аппаратному и программному обеспечению.

Убедитесь, что ваше мобильное устройство отвечает следующим аппаратным и программным требованиям:

- Версия Android от 5.0 и новее;
- Оперативная память не менее 256 Мб;
- Сенсорный экран;

## 2. Запуск мобильного приложения

Для запуска мобильного приложения Graph необходимо найти иконку приложения на рабочем столе вашего устройства и нажать на нее, если иконка отсутствует, перейдите в меню приложений, найдите нужное приложение в списке и нажмите на него. Если приложение отсутствует в меню приложений, попробуйте установить его повторно.

## 3. Интерфейс

После открытия Graph, перед вами должно открыться главное окно приложения. Оно содержит центральную область вывода графика, светлое поле в серую клетку и осью абсцисс и ординат, панель в нижней части экрана, которая предназначена для ввода коэффициентов в указанную в ней формулу, кнопку «Изменить» для подтверждения изменения формулы, панель в верхней части экрана, в которой указано название приложения и иконка меню.

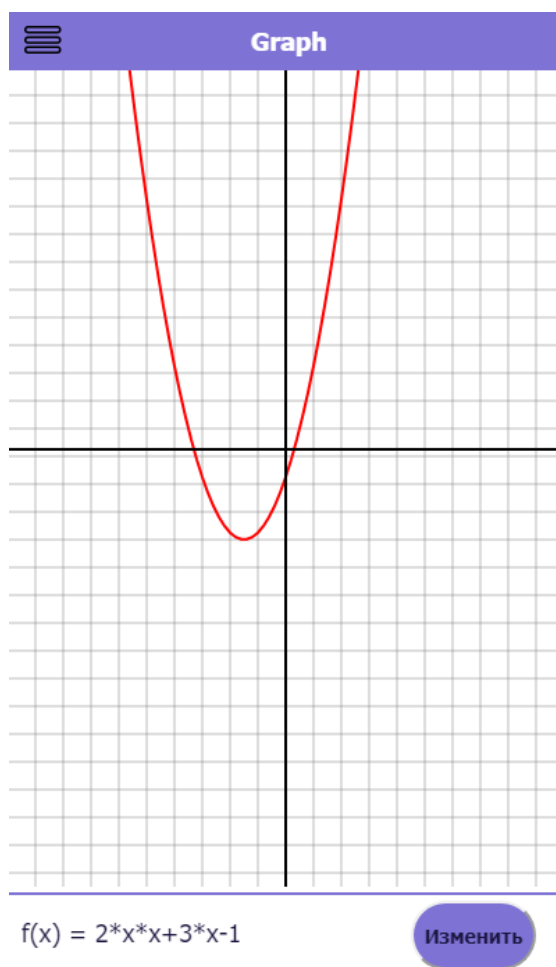


Рис 16. Главное окно приложения



При нажатии на кнопку меню, в левой части экрана приложения появится панель меню со списком доступных формул.

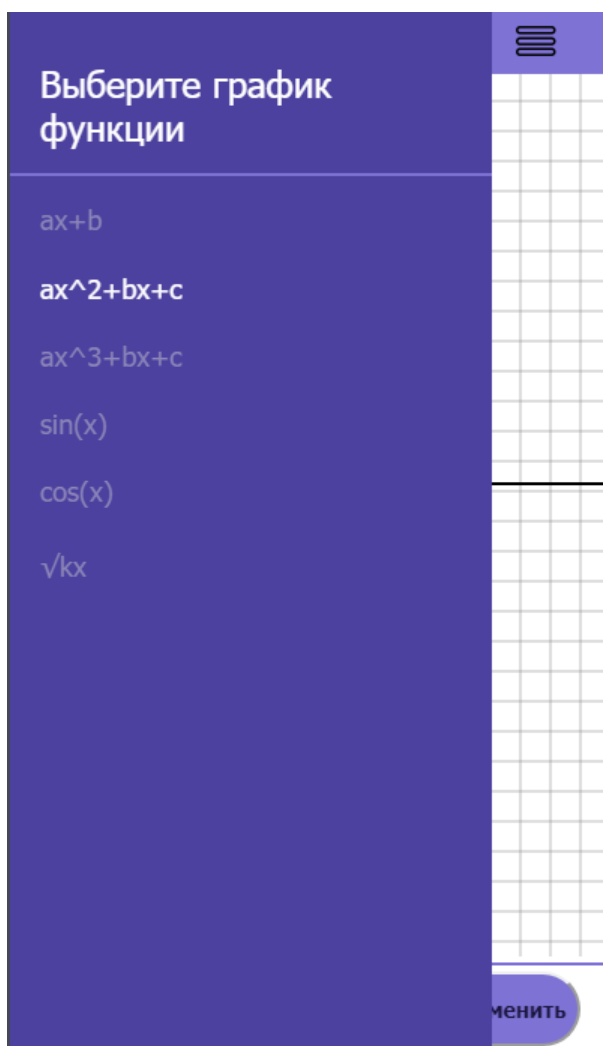


Рис 17. Панель меню доступных формул

#### 4. Работа с приложением

Для выбора необходимой формулы и работы с ней, необходимо нажать на иконку меню формул в верхней левой части экрана приложения. После открытия меню, выберите формулу и нажмите на нее, после чего формула должна выглядеть светлее по цвету, чем все остальные.

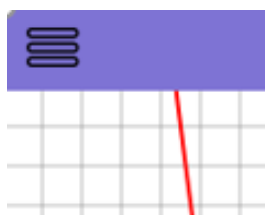


Рис 18. Иконка меню формул

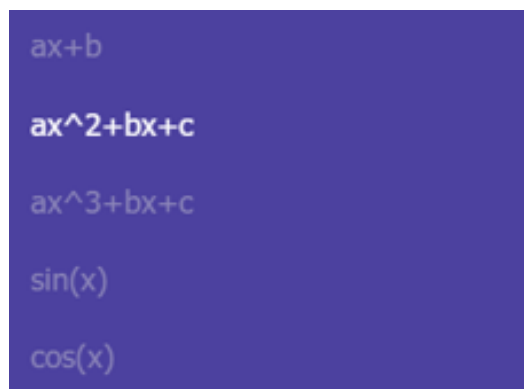


Рис 19. Пример цвета выбранной формулы

Закрываем меню нажав на ту же иконку вверху экрана, снова попадая в главное окно с осями координат, в нижней панели должна появиться выбранная формула, если она не совпадает с указанной вами, повторите предыдущие шаги, если формула совпадает, необходимо нажать на панель формул и на появившейся клавиатуре в доступные поля ввода коэффициентов ввести целые числа. Для подтверждения изменения формулы нажмите на кнопку «Изменить». Через пару секунд поле с осями координат обновится и построится график указанной вами функции.

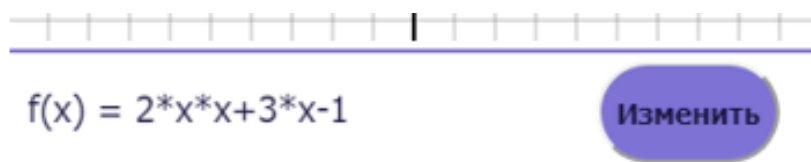


Рис 20. Панель с введенной формулой и кнопкой подтверждения

Для изменения выбранного графика, повторно коснитесь панели редактирования формулы, введите необходимые целые числа и подтвердите изменения. Если вы хотите поменять график функции, повторите пройденные шаги сначала (см. пункт 4).

## 5. Правила изменения формул в панели редактирования

На примере формулы  $f(x) = 2*x*x+3*x-1$ , рассмотрим особенности изменения вида конкретной формулы и изменения коэффициентов.

Данная формула представляет собой параболу с числовыми коэффициентами 2, 3 и 1, чтобы добиться смещения графика вправо, влево,

вверх или вниз, необходимо изменить коэффициенты, как и в обычной математике, например, нам нужно центровать график на начале координат по  $x$  и  $y$ , вводим коэффициенты 2, 1, 0, и мы получаем нужный нам результат.

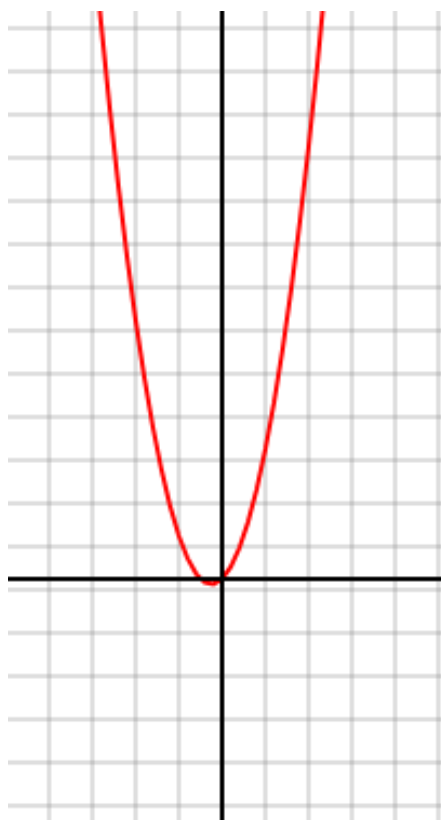


Рис 21. Пример графика  $f(x) = 2x^2 + x$

Если нас интересует обычная прямая  $f(x) = x$ , вводим коэффициенты 0, 1, 1 в формулу  $f(x) = 0 \cdot x^2 + 1 \cdot x - 0$ , и получаем нужный нам график.

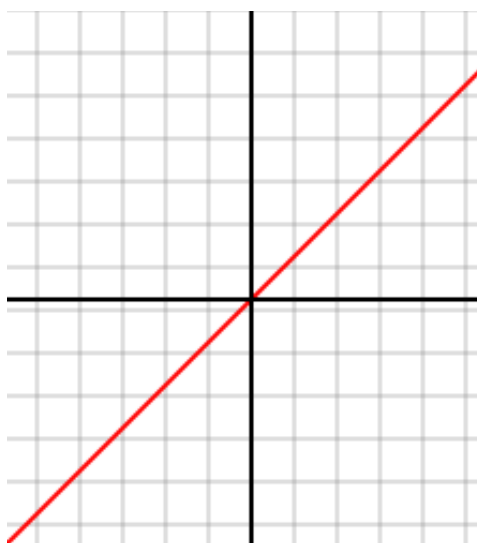


Рис 22. Пример графика  $f(x) = 0 \cdot x^2 + 1 \cdot x - 0$

Таким образом, мы рассмотрели пример изменения изначальной функции  $f(x) = 2x^2 + 3x - 1$  для получения более простой функции  $f(x) = x$ , изменяя лишь ее коэффициенты.

## **Заключение**

Для выпускной квалификационной работы было решено, используя элемент HTML Canvas, написать мобильное приложение для построения простейших графиков функций.

При анализе материала, относящегося к мобильным приложениям, была выявлена проблема внедрения мобильных устройств в образовательный процесс школ, колледжей и институтов, были рассмотрены достоинства и недостатки использования мобильных технологий. В результате исследования основных проблем темы, была доказана актуальность разработки обучающего мобильного приложения.

Перед началом разработки были составлены IDEF0 и UML диаграммы для модельного представления разрабатываемого проекта, написано техническое задание и представлен дизайн интерфейса приложения.

В результате написания выпускной квалификационной работы было разработано мобильное приложение для построения графиков функций в соответствии с техническим заданием.

## Список информационных источников

1. Сколько времени и какими приложениями пользуются россияне // AppTractor - медиа о разработке и продвижении новых IT-проектов. URL: <https://apptractor.ru/info/analytics/skolko-vremeni-i-kakimi-prilozheniyami-polzuyutsya-rossiyane.html> (дата обращения: 16.04.2020).
2. Абдураззакова Дильдора Анваровна. Мобильные технологии в образовательном процессе // Наука и образование сегодня. 2019. №2 (37). URL: <https://cyberleninka.ru/article/n/mobilnye-tehnologii-v-obrazovatelnom-protssesse> (дата обращения: 16.04.2020).
3. Шарифбаева Халида Ядкарровна. Применение мобильных технологий в сфере образования // Проблемы педагогики. 2018. №2 (34). URL: <https://cyberleninka.ru/article/n/primenenie-mobilnyh-tehnologiy-v-sfere-obrazovaniya> (дата обращения: 16.04.2020).
4. РЕКОМЕНДАЦИИ ПО ПОЛИТИКЕ В ОБЛАСТИ МОБИЛЬНОГО ОБУЧЕНИЯ // Информационно-публицистический образовательный журнал "ИНТЕРАКТИВНОЕ ОБРАЗОВАНИЕ" URL: <https://interactiv.su/2020/04/03/рекомендации-по-политике-в-области-мо/> (дата обращения: 16.04.2020).
5. Чернявский Борис Андреевич. Самообразование с помощью мобильных приложений // Вопросы науки и образования. 2019. №23 (71). URL: <https://cyberleninka.ru/article/n/samoobrazovanie-s-pomoschyu-mobilnyh-prilozheniy> (дата обращения: 17.04.2020).
6. Шарифбаева Халида Ядкарровна, Абдураззакова Дильдора Анваровна. Формы и методы мобильного обучения // Достижения науки и образования. 2018. №13 (35). URL: <https://cyberleninka.ru/article/n/formy-i-metody-mobilnogo-obucheniya> (дата обращения: 17.04.2020).
7. Доскажанов Ч.Т., Даненова Г.Т., Коккоз М.М. РОЛЬ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ В СИСТЕМЕ ОБРАЗОВАНИЯ // Международный журнал

- экспериментального образования. – 2018. – № 2. – С. 17-22; URL: <http://expeducation.ru/ru/article/view?id=11790> (дата обращения: 17.04.2020).
8. Толстоухова Ирина Васильевна Мобильные информационно-коммуникативные технологии обучения в профессиональной подготовке инженеров // Вестник ТГПУ. 2016. №9 (174). URL: <https://cyberleninka.ru/article/n/mobilnye-informatsionno-kommunikativnye-tehnologii-obucheniya-v-professionalnoy-podgotovke-inzhenerov> (дата обращения: 19.04.2020).
9. Стандарт ТСО // Справочник URL: [https://spravochnik.ru/standartizaciya/standart\\_tco/#standarty-tco-92-i-tso-95](https://spravochnik.ru/standartizaciya/standart_tco/#standarty-tco-92-i-tso-95) (дата обращения: 19.04.2020).
10. Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.2.2821-10 «Санитарно-эпидемиологические требования к условиям и организации обучения в общеобразовательных организациях» от 25 декабря 2013 г. N 72. М.: Роспотребнадзор. Введен в действие с 1 сентября 2011 г. Изменения и дополнения от: 29 июня 2011 г., 25 декабря 2013 г., 24 ноября 2015 г.
11. Абдрашева Г.К., Туткышбаева Ш.С., Калибекова Д.Ш. Мобильное обучение и мобильные приложения в образовании // Проблемы и перспективы развития образования в России. 2016. №39. URL: <https://cyberleninka.ru/article/n/mobilnoe-obuchenie-i-mobilnye-prilozheniya-v-obrazovanii> (дата обращения: 19.04.2020).
12. Статистика операционных систем в мире // Marketer — интернет издание про маркетинг и технологии для бизнеса, стартапов и предпринимателей. URL: <https://marketer.ua/stats-operating-system-2020/> (дата обращения: 19.04.2020).
13. Android URL: <https://www.android.com> (дата обращения: 19.04.2020).
14. Как работать с Play Console // Справка - Play Console URL: <https://support.google.com/googleplay/android-developer/answer/6112435?hl=ru> (дата обращения: 20.04.2020).

- 15.App Store URL: <https://www.apple.com/ru/ios/app-store/principles-practices/>  
(дата обращения: 20.04.2020).
- 16.Android или iOS: что выбрать для старта карьеры в мобильной разработке // Rusbase URL: <https://rb.ru/opinion/android-ili-ios/> (дата обращения: 20.04.2020).
- 17.Мобильные приложения: нативные, веб и гибридные // agile - услуги по разработке веб и мобильных приложений URL: <https://agilie.com/ru/blog/mobilnyie-prilozhieniia-nativnyie-vieb-i-ghibridnyie>  
(дата обращения: 20.04.2020).
- 18.Разработка под iOS и Android: рейтинг языков программирования 2020 // AppTractor - медиа о разработке и продвижении новых IT-проектов. URL: <https://apptractor.ru/rejting-yazykov-programmirovaniya-2020> (дата обращения: 21.04.2020).
- 19.Swift URL: <https://www.apple.com/ru/swift/> (дата обращения: 21.04.2020).
- 20.Руководство по языку Kotlin // Kotlin URL: <https://kotlinlang.ru> (дата обращения: 22.04.2020).
- 21.Dart // Википедия. Свободная энциклопедия URL: <https://ru.wikipedia.org/wiki/Dart> (дата обращения: 22.04.2020).
- 22.JavaScript // Википедия. Свободная энциклопедия URL: <https://ru.wikipedia.org/wiki/JavaScript> (дата обращения: 22.04.2020).
- 23.Подробнее о технологии Java // Java URL: <https://www.java.com/ru/about/>  
(дата обращения: 22.04.2020).
- 24.Фреймворк // Википедия. Свободная энциклопедия URL: <https://ru.wikipedia.org/wiki/Фреймворк> (дата обращения: 23.04.2020).
- 25.Один фреймворк. Mobile и desktop. // Angular URL: <https://angular-ru.github.io>  
(дата обращения: 23.04.2020).
- 26.React Native // Википедия. Свободная энциклопедия URL: [https://en.wikipedia.org/wiki/React\\_Native](https://en.wikipedia.org/wiki/React_Native) (дата обращения: 23.04.2020).



- 27.PhoneGap // Википедия. Свободная энциклопедия URL: <https://ru.wikipedia.org/wiki/PhoneGap> (дата обращения: 23.04.2020).
- 28.РАЗРАБОТКА МОБИЛЬНЫХ ПРИЛОЖЕНИЙ НА IONIC FRAMEWORK // ionic URL: <https://www.ionic-framework.ru> (дата обращения: 24.04.2020).
- 29.Microsoft Visual Studio URL: <https://visualstudio.microsoft.com/ru/> (дата обращения: 24.04.2020).
- 30.PyCharm URL: <https://www.jetbrains.com/ru-ru/pycharm/> (дата обращения: 24.04.2020).
- 31.Eclipse (среда разработки) // Википедия. Свободная энциклопедия URL: [https://ru.wikipedia.org/wiki/Eclipse\\_\(среда\\_разработки\)](https://ru.wikipedia.org/wiki/Eclipse_(среда_разработки)) (дата обращения: 25.04.2020).
- 32.Android Studio // Google Developers URL: <https://developer.android.com/studio> (дата обращения: 25.04.2020).
- 33.Web Frameworks: How To Get Started //.djangostars URL: <https://djangostars.com/blog/what-is-a-web-framework/> (дата обращения: 26.04.2020).
- 34.Get Started Fast // Apache Cordova URL: <https://cordova.apache.org> (дата обращения: 26.04.2020).

## **Приложения**

### **Приложение 1.**